Status Review
MUGSy Design
July 19, 1996

MUGSy Design Team

- Overview: Darlene West
- Design/Development Environment: Steve Robinson
- CM: Yuri Frankel and Paul Capotosto
- CRM: Gary Orr
- DAT: Ed Burgess
- Product Management/Publication: Maria Jacob and June Phillips
- Integration: Larry Barrett
- Tool I/F: Steve Robinson

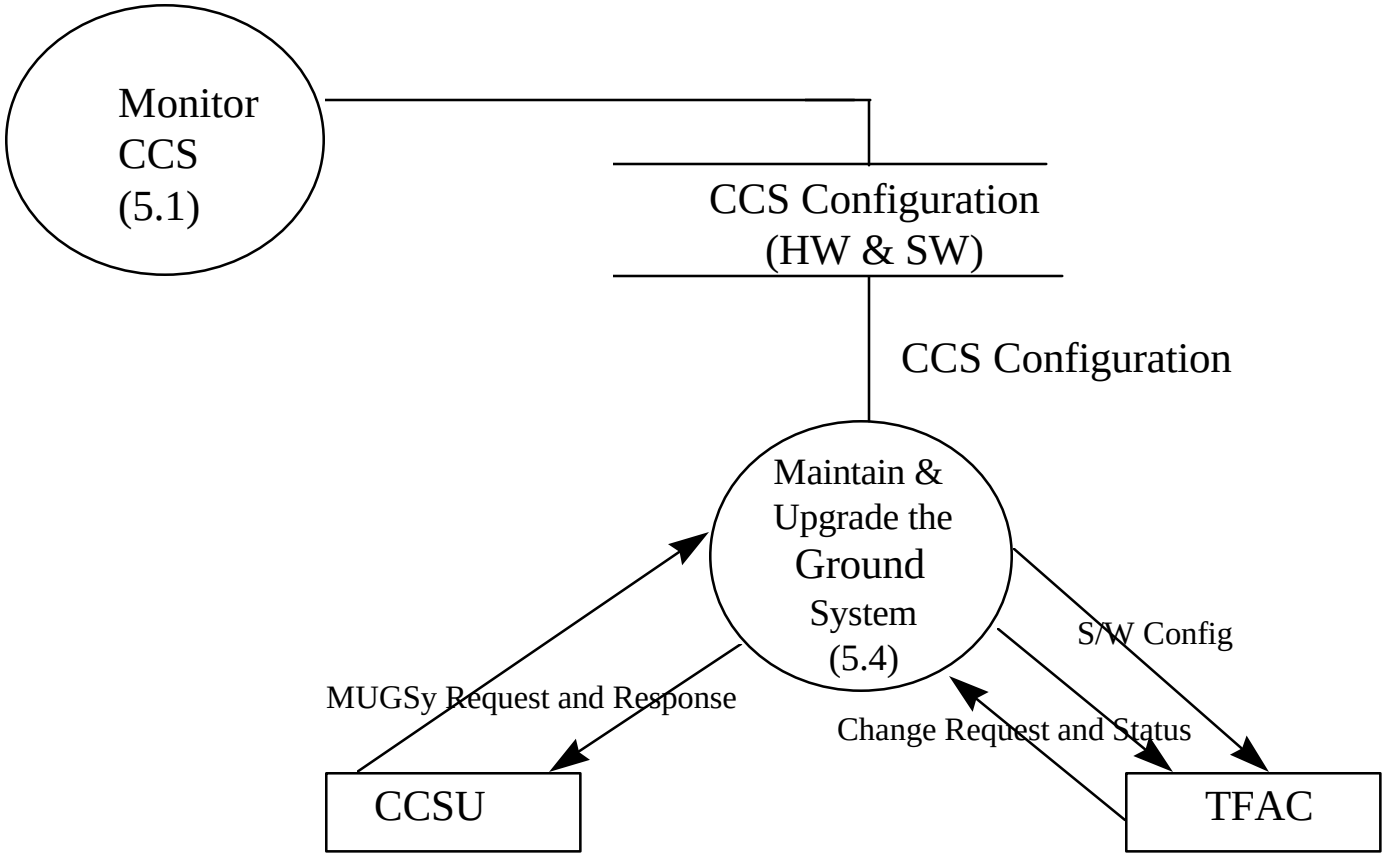
MUGSy Design Agenda

- What is MUGSy??
  - MUGSy Capabilities
  - MUGSy Context Diagram
  - MUGSy Conceptual Design (Changes)
- What MUGSy means to CCS and VISION
  - MUGSy User Types
  - MUGSy Logical Architecture
  - CCS Engineering Model
  - Repository Structure and User Types for Selected Products
  - MUGSy Physical Architecture
- What's next for MUGSy

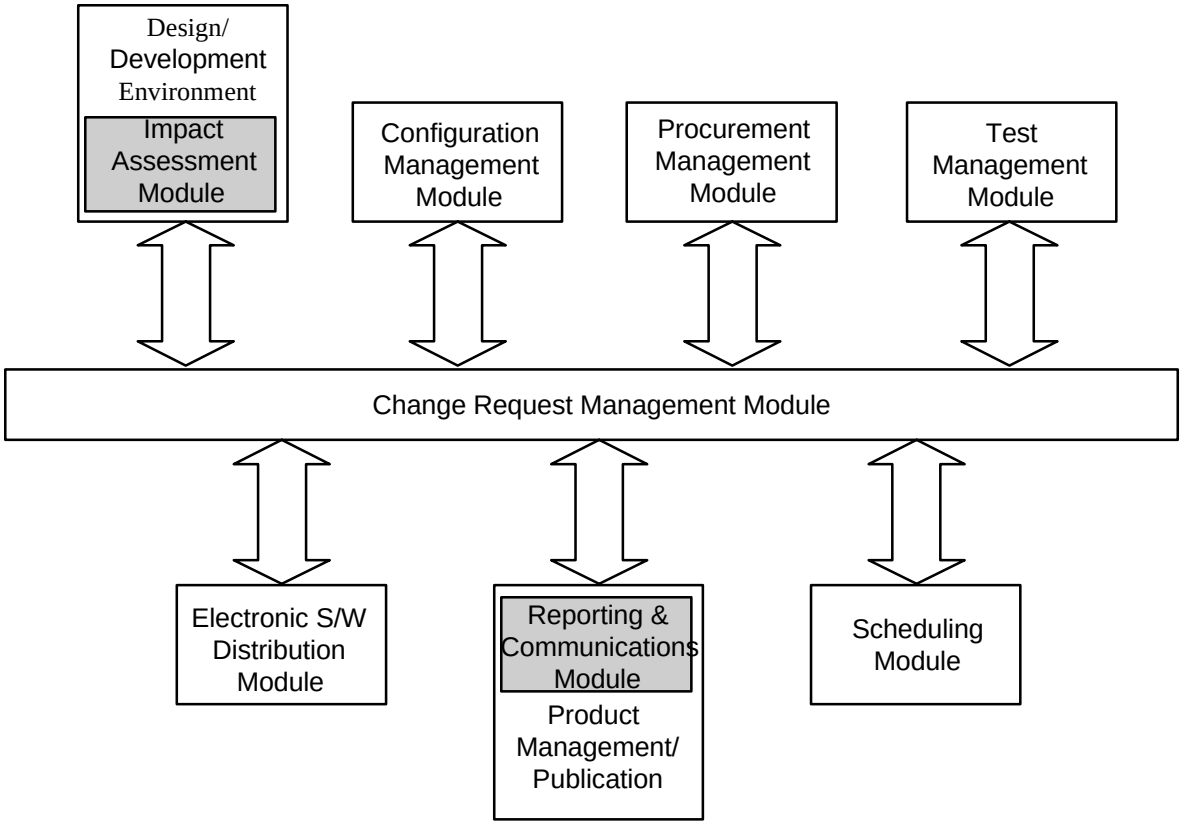
What is MUGSy??: An integrated development and maintenance system with the following Capabilities (**new**)

- **Provides the CCS Design and Development Environment**
- **Provides the capability to manage product generation/modification process and distribution**
- Provides the capability to manage the change process for changes to the CCS design and development
- Provides the capability to manage CCS design, development, product and test configurations
- Provides the test environment

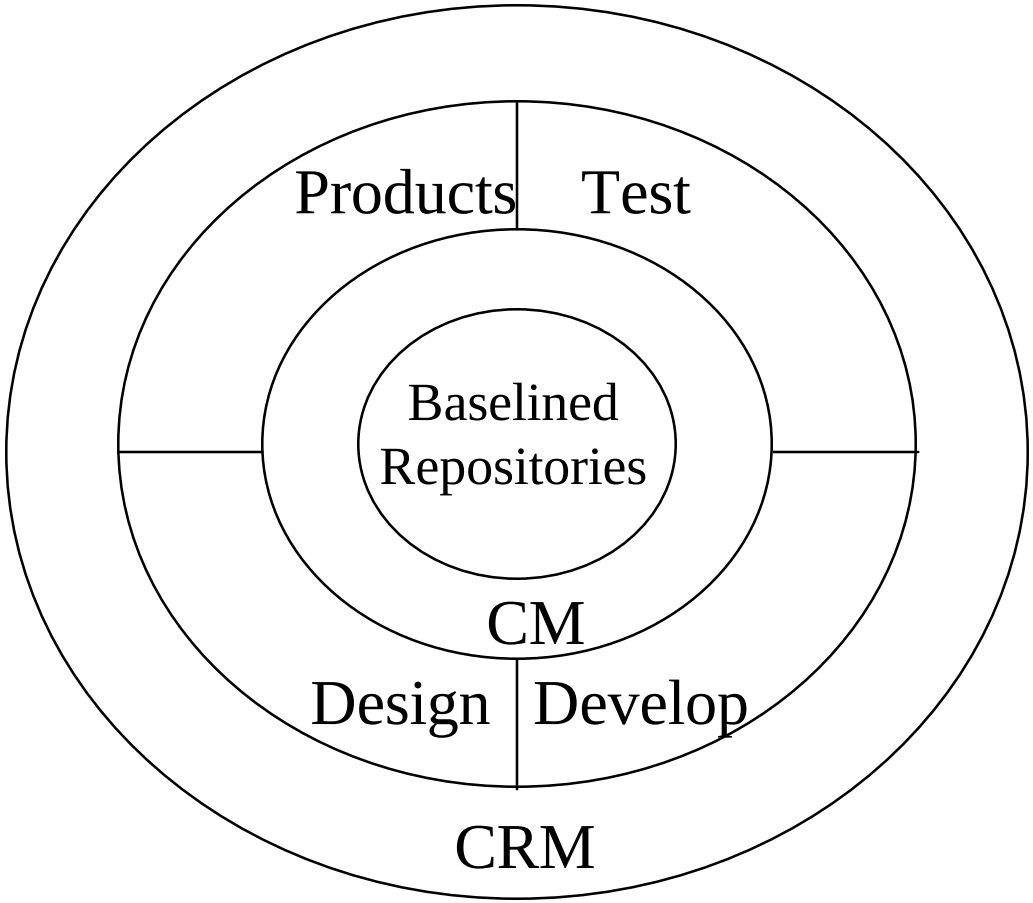
MUGSy Context Diagram



Conceptual Design: Core Modules



Conceptual Design: Product Relationships



What MUGSy means to CCS and VISION

- MUGSy User Types
- MUGSy Logical Architecture
- Engineering Model
- Products Selected (to date) to “Implement MUGSy”

MUGSy User Roles

	C R U D		Tools	Candidate
Information Engineers	C R U	ERD, Data Dictionary, Physical Database	Database Design tools	Designer 2000
System Engineers*	C R U	System Design, Rqmts, Threads, System	Upper Case, Text Editor, Performance Modeling Tool	System Architect, Designer 2000
Software Engineer	C R U	S/W Design, Code, SDF, Legacy Code, COTS	Lower Case, Compilers, Linkers, Debuggers, Text Editor, Code Converters	System Architect
Test Engineers	C R U	Inputs, Threads, Test Metrics, Eval Criteria, Test Pre-requisites	Path Coverage, Test Management, Key Capture/Playback, Editor, Load Testing, SQL Testing	Purify, Mercury, XRunner, SATAN

MUGSy User Roles (Cont'd)

	C R U D		Tools	Candidate
	C R U	Status,	Generator, Text Editor. Client/Server Test Tools, SQL testing	MS Project
Operations (Reviewers)	C R U	Operational Procedures (PSTOLS), Expert System Rules/Models	Text Editor	MS Access
CM Manager	C R U D	Repository Structure	CM Tool, CRM Tool, Baselines	Aide-de-Camp, SCOPUS
	R U	*CR Database*	*Generator*	
CR Assignee	*R U*	*Schedule,*	*Scheduling, CRM Generator*	

MUGSy User Roles (Cont'd)

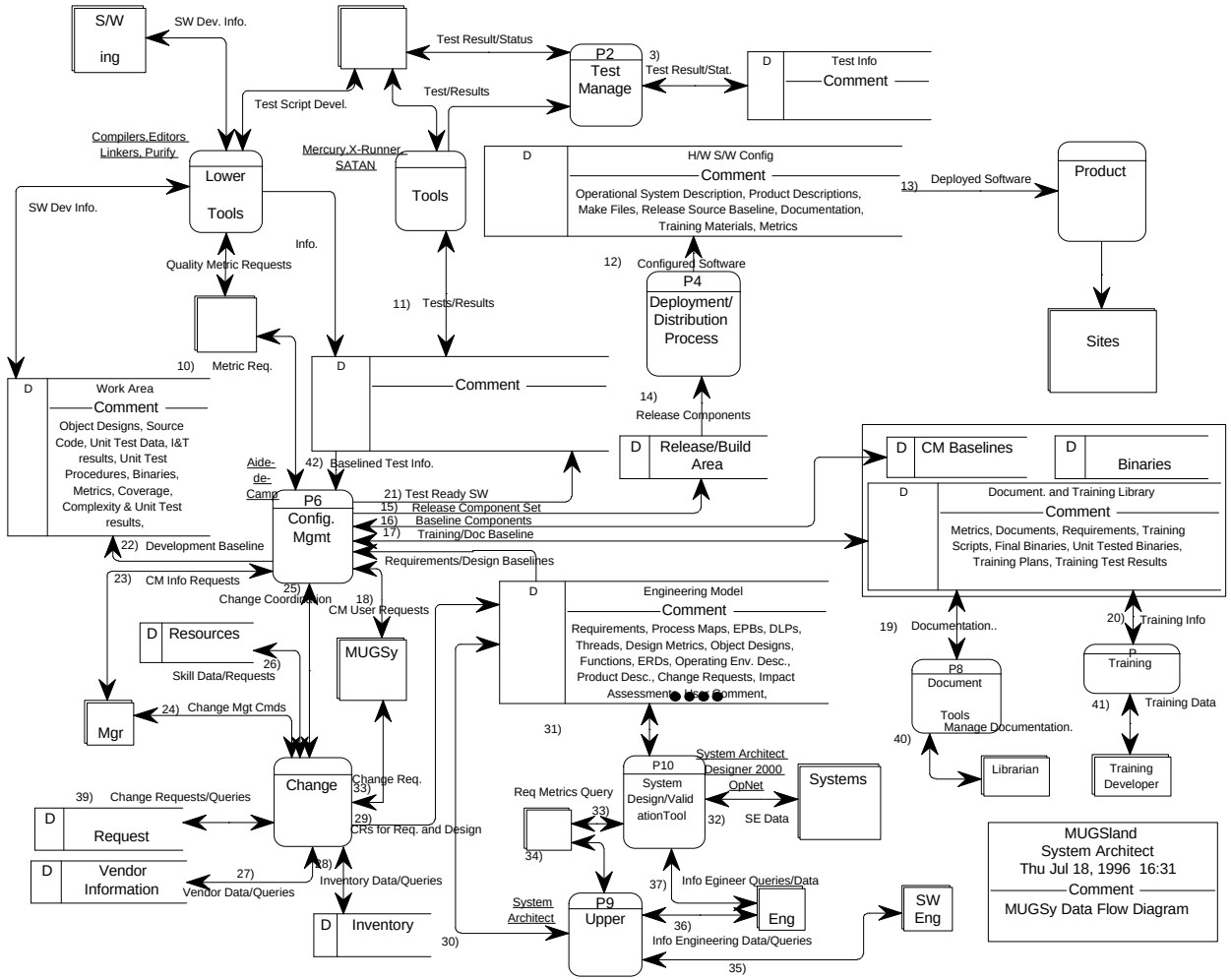
CCS USER TYPES	C R U D		Tools	Candidate
Logistics Manager	C R U D	Inventory, Procurement	CRM Tool	SCOPUS, "RDBMS"
QA	C R	Information	Metric Collection Tools	
Librarian	R U D	Web pages, products	Formatter, Editor, Document Pubs	
MUGSy Admin	C R U D	Administrative, configuration	"Admin" tools	
CR Originator	*C R U*		*CRM Tool*	*SCOPUS*

MUGSy User Roles (Complete)

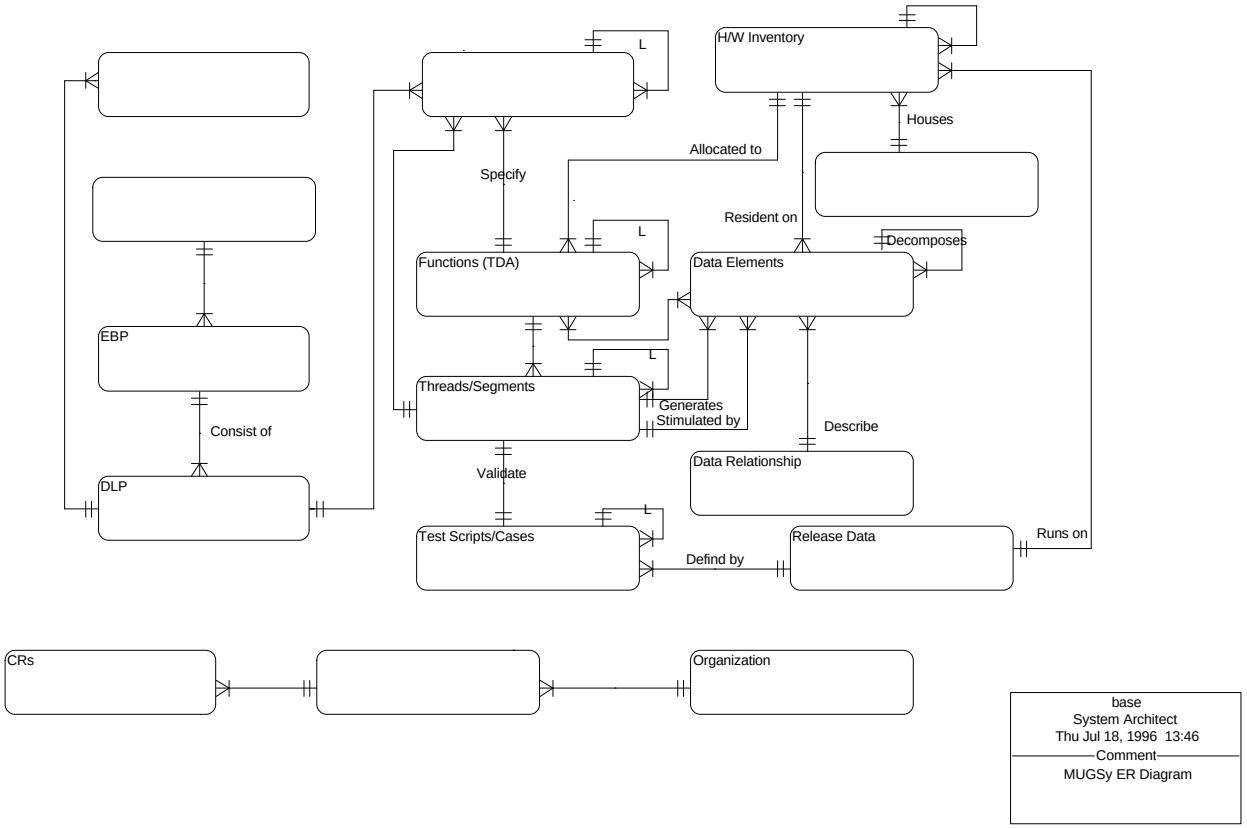
CCS USER TYPES	C R U D		Tools	Candidate
Training Manager	C R U	Training Materials	Authoring Tools	
Integration Manager	C R U	Release	Database & Report Generator	

* Includes Network &

Logical Architecture



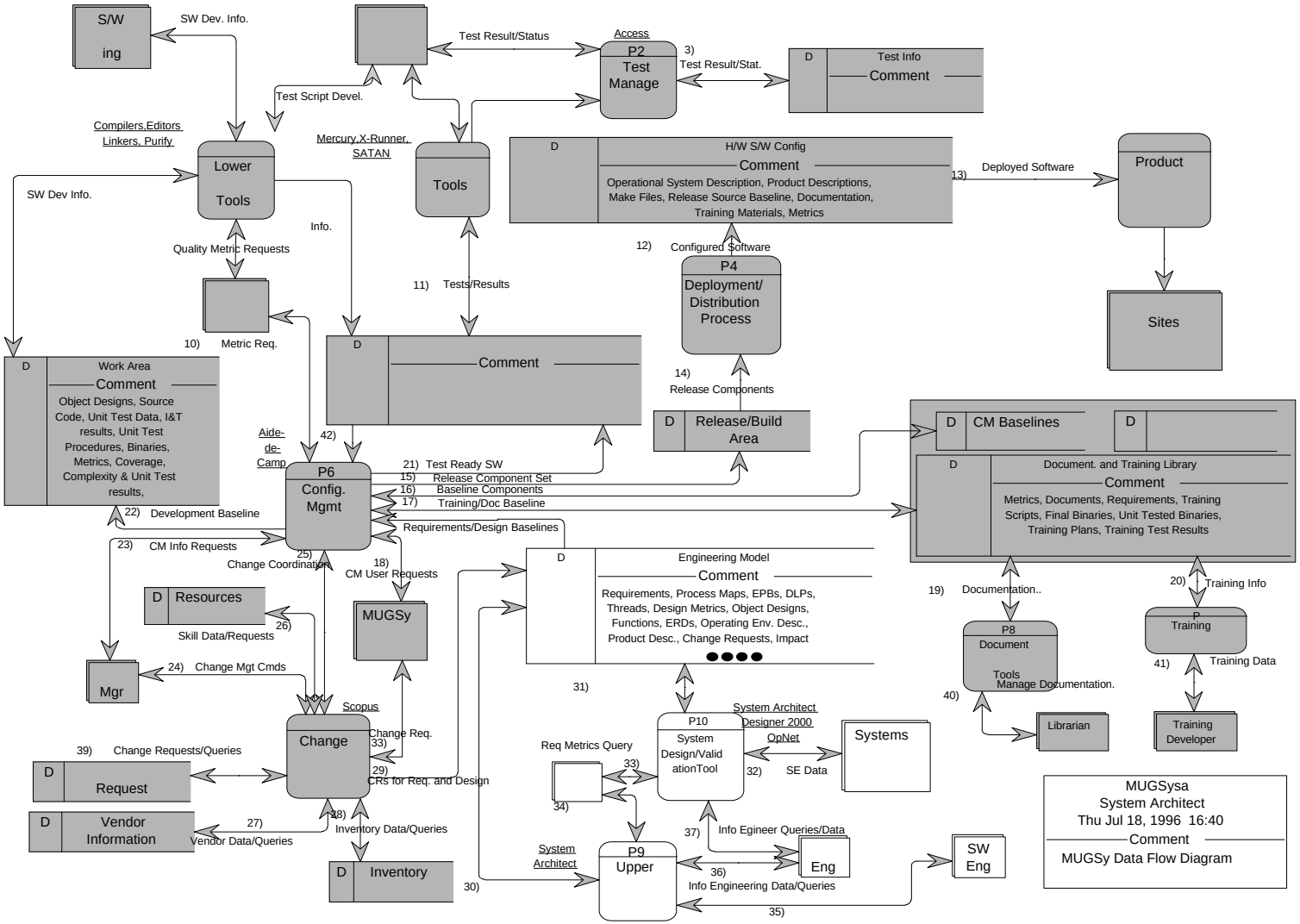
CCS Engineering Model



Products selected to “implement MUGSy” (**Covered in presentation**)

- **System Architect**
- **Aide-de-Camp**
- **SCOPUS**
- MS Office
- Other Products: Designer 2000, Test tools

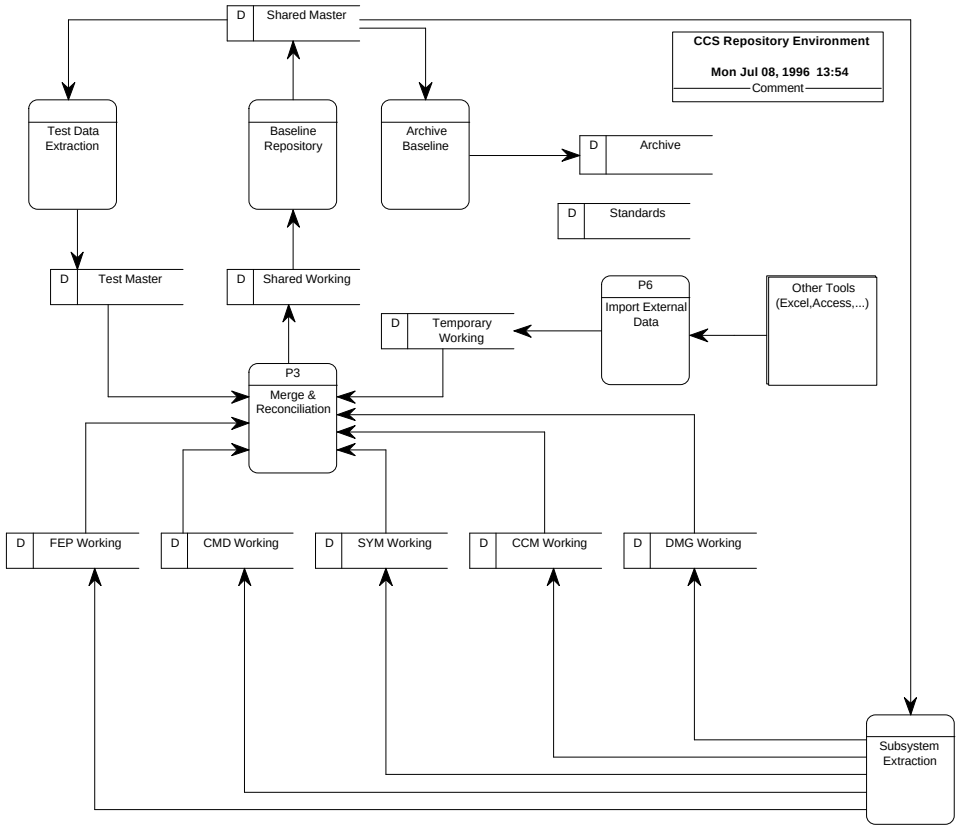
Capabilities provided by System Architect



System Architect Supports Designers, Developers, and Managers

- Provides a single, diagram-oriented, interface to capture and distribute all CCS functional design information
- Performs automated design validation functions including interface validation and design completeness checks
- Provides a method of allocating requirements to the various software components
- Provides a large variety of canned reports that can be used for walkthroughs and reviews
- Tailorable to support specific CCS information capture needs

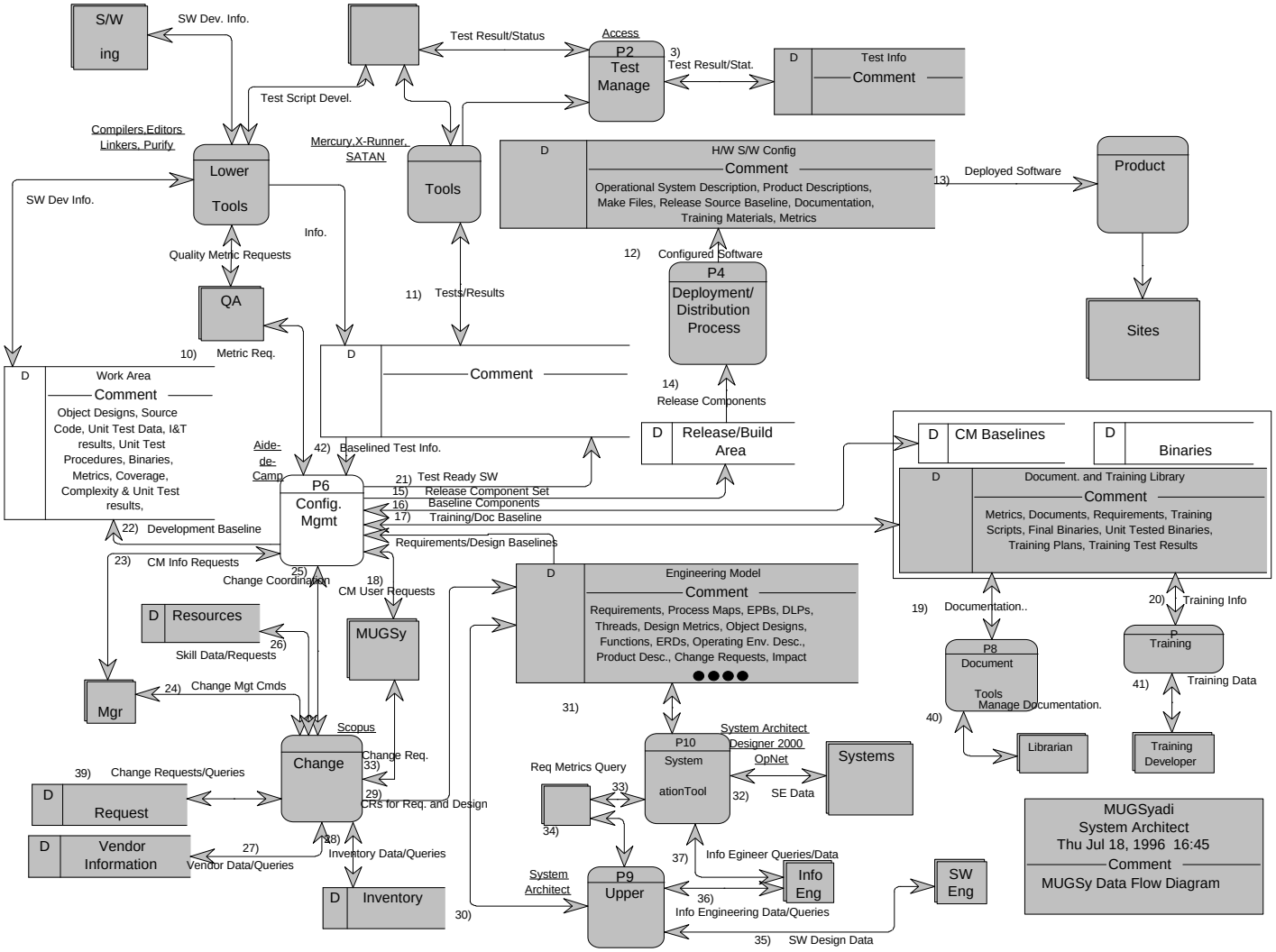
System Architect Repository Structure



System Architect User Roles

- System Engineers - define TDA, system-level design, requirements
- Software Engineers - define software design (object diagrams, structure charts)
- Project Manager - generate design reports, metrics reports
- Quality Assurance Manager - perform audits, calculate metrics
- Test Engineers - manage thread definitions
- MUGSy Administrator - define processes and roles, perform administrative functions, manage and merge encyclopedias, maintain baselines

Capabilities provided by Aide-de-Camp



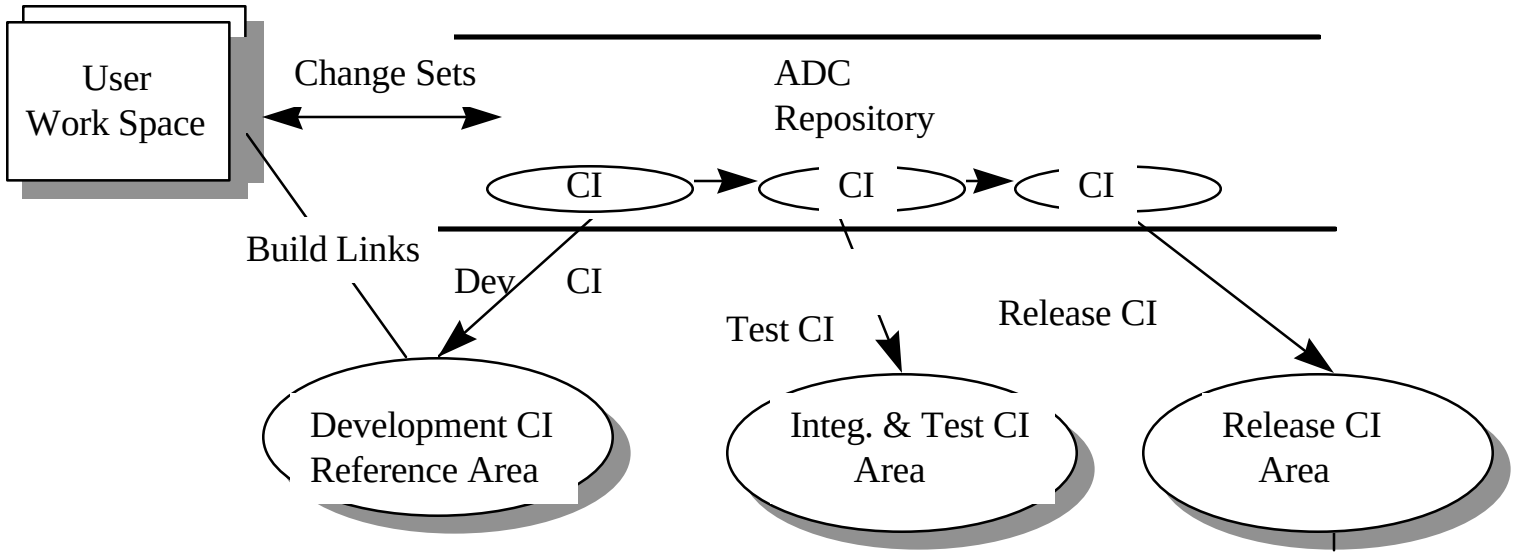
Aide-de-Camp Supports Designers, Developers, and Managers

- Creates and maintains file repositories and Configuration Items (CI)
- Automates repositories and CI version versus file version
- Supports parallel and concurrent development
- Operates check-in, check-out on a group of files via Change Sets
- Performs branching, migrating, merging and removing Change Sets
- Automates conflict detection and annotation in Change Set merging

Aide-de-Camp Supports Designers, Developers, and Managers

- Provides graphical version trees and access to any entity and entity release and version
- Provides scanners to capture dependencies within C, C++, FORTRAN, ADA code
- Provides release and software audits with full list of files and relationships
- Provides entity and summary reports (name, date, changes, descriptions, etc.)

CI Promotion and Release Process



CCS Structure and Naming Conventions for repositories and reference areas

ADC Repository Structure /adcdisk/adc/

- /adccnf
- /adcrep
  - /adcpro
  - /scopus
  - /GUI
  - / mdlware
  - /isis
  - /isp
  - /red-brick
  - /tlm-proc
  - /DB-support
  - /pv-wave
  - /ultrix53
  - /ultrix62
  - /unix
  - /win-nt

Reference Area Structure /adcdisk/adc/

- /adcrefa
  - /GUI
  - /mdlware
  - /isis
  - /isp
- /cnfa

Aide-de-Camp Repository Structure for CCS Custom Code

File structure and naming convention for a CCS custom code development:

/root-name/

- /src
  - /c
  - /cpp
  - /for
- /inc
  - /incc
  - /inccpp
  - /incfor
- /lib
- /doc
- /test
- /cnf
- /obj
- /exe

Directories content description

- src - source files for a one language code or a source file subdirectories to separate source code of two or more languages
  - c - C source files
  - cpp - C++ source files
  - for - FORTRAN source files
- inc - include files for a one language code or an include file subdirectories to separate include files or two or more languages
  - c - C include files
  - cpp - C++ include files
  - for - FORTRAN source files
- lib - local object libraries
- doc - documentation, journals and notes
- test - test data and reports
- cnf - hardware and software configuration files
- obj - object files (will NOT be included in repository)
- exe - executable files (will NOT be included in repository)

Aide-de-Camp Repository Structure for JAVA Code

File structure and naming convention for a CCS Java code development:

/root-name/

/src

/htm

/doc

/test

/classes

Directories content description

src - Java source files

htm - html files

doc - documentation, journals and notes

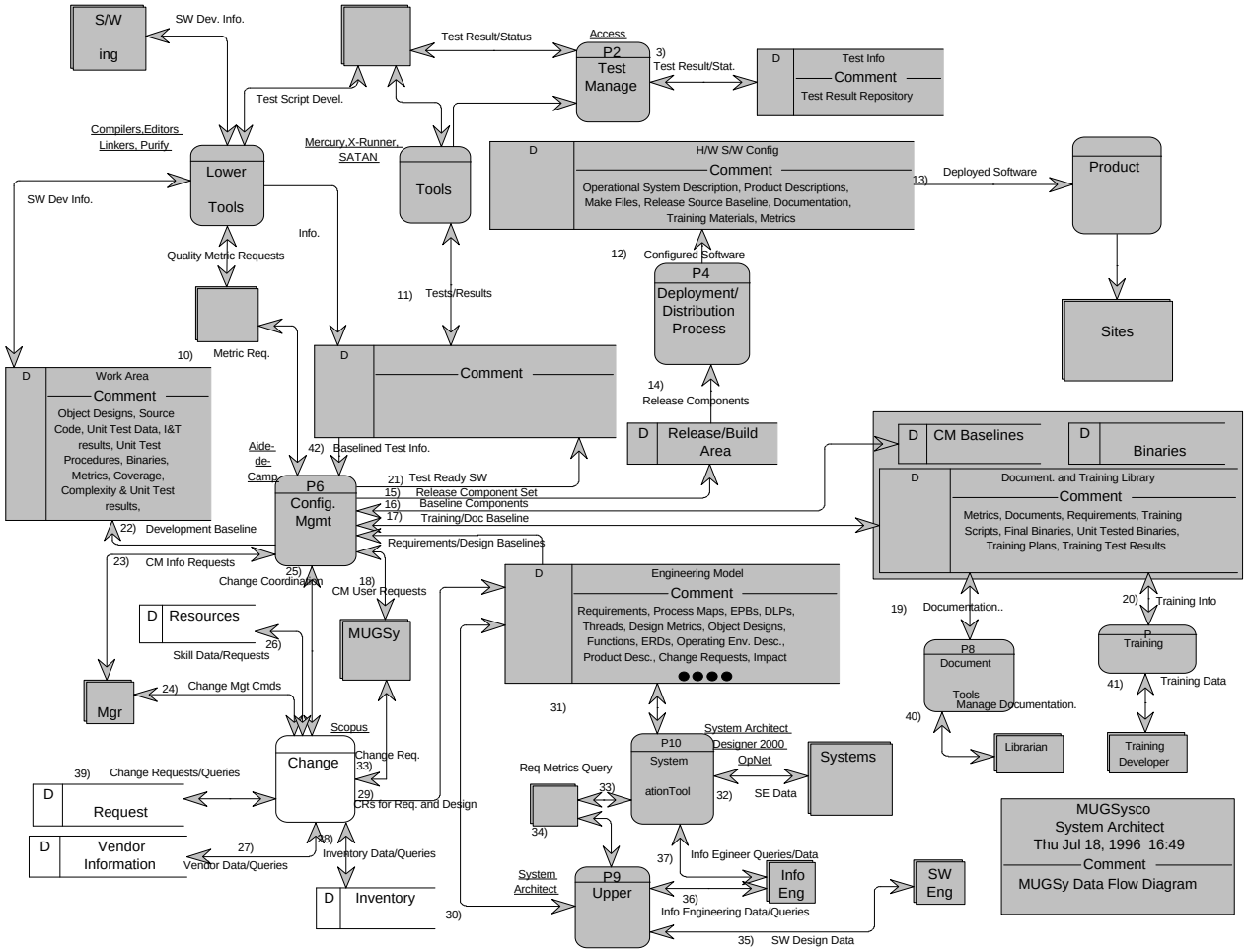
test - test data and reports

classes - Java binary files (will NOT be included in repository)

Aide-de-Camp User Roles

- CM Administrator - Maintain and administer ADC
- Developer - Manage Segment/repository file
- Task/Project Lead - Maintain repository/configuration
- Repository Administrator - Administer repositories
- Release Manager - Manage release configuration

Capabilities provided by SCOPUS



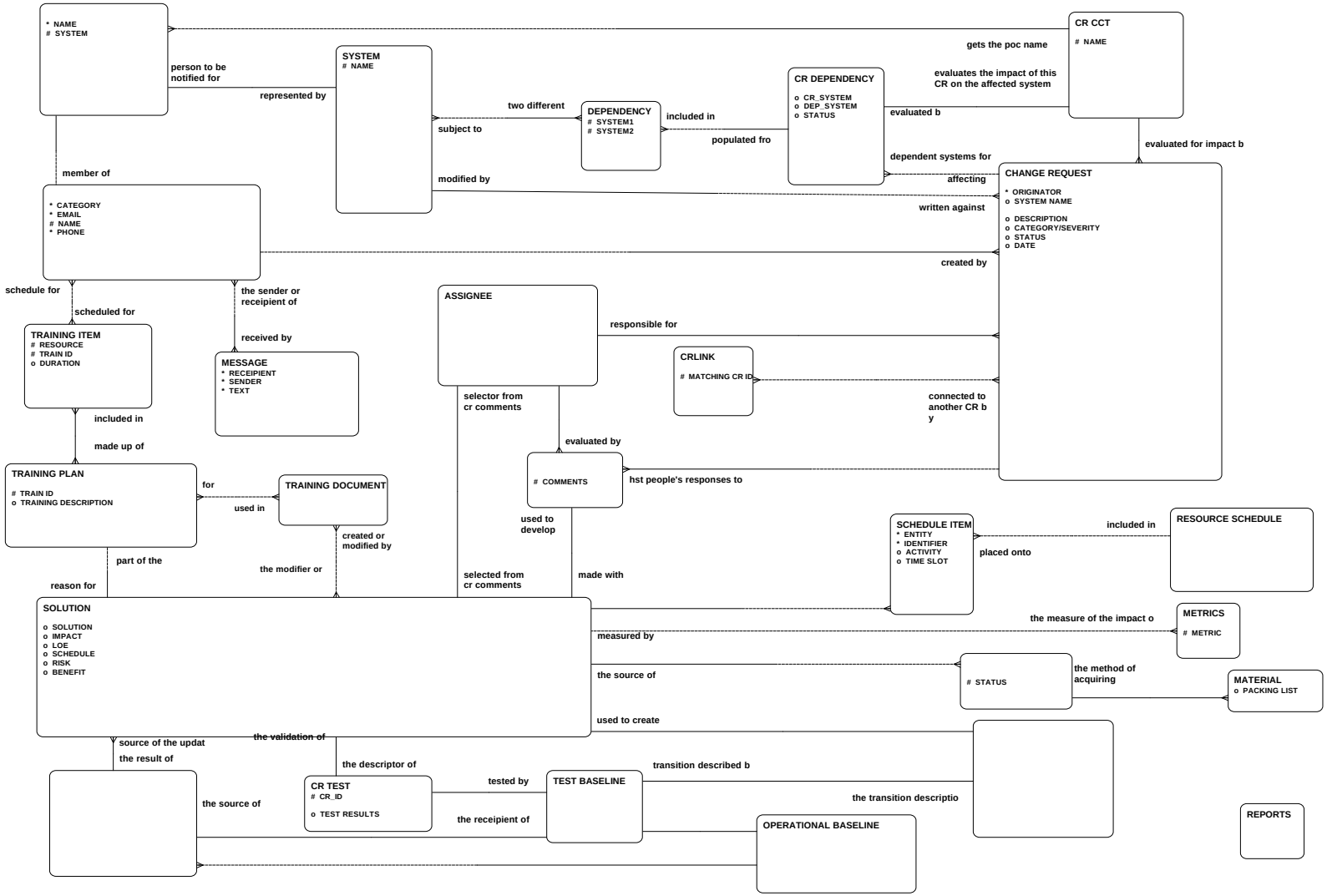
SCOPUS Supports Designers, Developers, and Managers

- Tracking
  - Change Requests (CR's)
  - Solutions
  - Products
    - Documentation
    - Hardware and Software
    - Configured Items
  - Resources & related skills
- Interface to the Web
  - Submit CR's
  - Query the SCOPUS database

SCOPUS Supports Designers, Developers, and Managers

- Uses “Business Rules” (defined by Condition that causes a rule to be applied) for workflow management and automation
  - Notification
    - Fax
    - E-mail
    - Pager
  - Escalation
    - Execute a procedure
    - Change a field value in the table
  - Generate reports automatically based on time

SCOPUS Repository Structure



SCOPUS User Roles

- CR Originator - Submit and query CR's
- Logistics Manager- Track Purchase Orders, Hardware/Software inventory, Vendor Information
- Project Manager - Track personnel and related skills
- CR Coordinator - Track CR's
- Test Engineer - Track Test Results
- CM Manager - Track Configuration Items
- Librarian - Generate Manager Reports

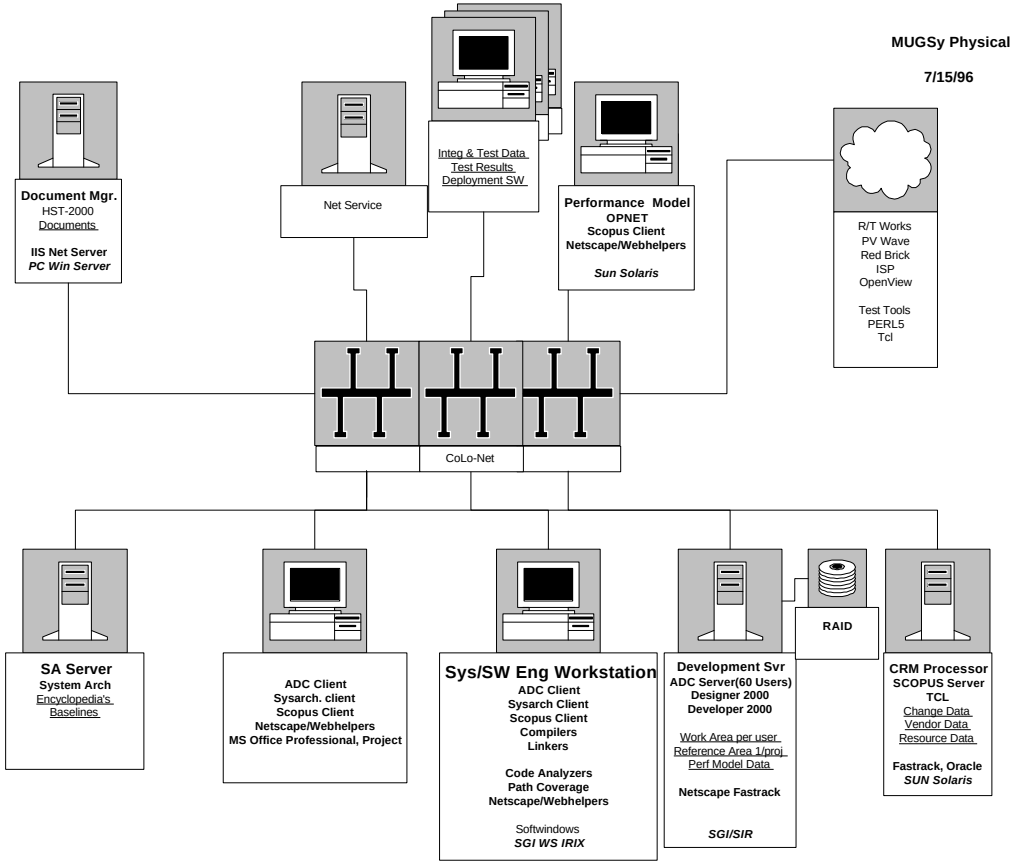
Product Generation/Modification Repository Structure

- Formal Products
  - Uses V2K BDC1 Server
  - Directories Defined for
    - Product Development
    - Product Review
    - Product Editing
    - Product Baseline
  - Baselined Products
    - Published on CCS Web Page in Library
- Informal Products published on CCS Web Page

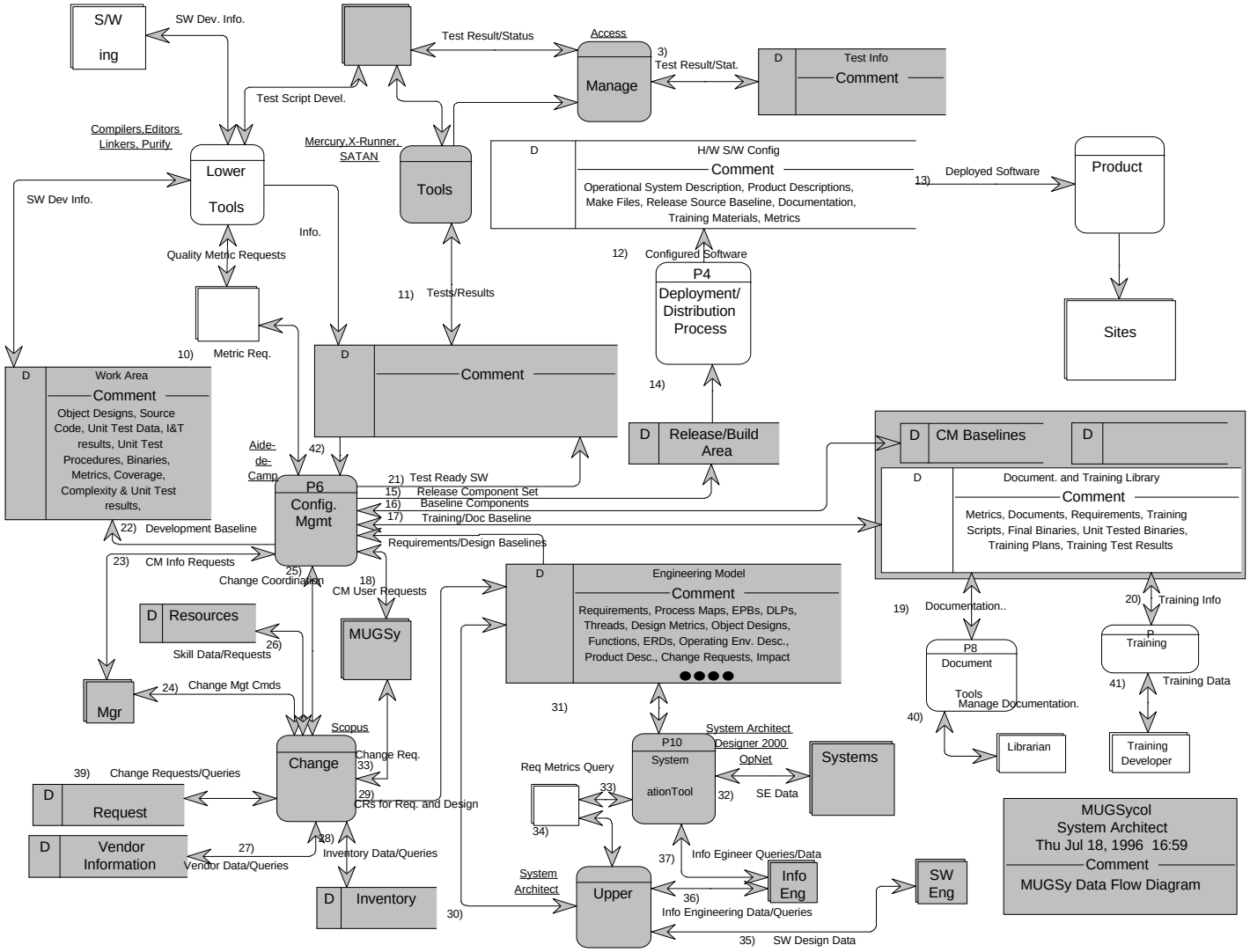
Product Generation/Modification User Types

- Librarian - Manages process of generating, reviewing, editing, and baselining formal products
- Author - Generates, modifies, or edits formal product
- Reviewer - Reviews formal products
- Approver - Approves formal product
- WebMaster - Publishes informal products:

Physical Architecture



Products Coverage of the MUGSy Architecture



What's next for MUGSy to Complete Release 1?

- Select Test Tools
- Complete procurement and installation of each product
- Populate each product
- Customize each product
- Establish process/procedures for using products/"integrating products"
- Train Users
- Maintain and administer MUGSy

What's next for MUGSy Release 2 and after?

- Identify candidate procedures for automation
- Automate procedures with "payback"
- Integrate products
- Identify additional products to aid automation initiative
- Identical activities as for R1

Issues

- Lower Case Software Development Tools
  - Within MUGSy SCOPE ??
  - OUTSIDE of MUGSy??